Programmed stall cycles slow-down video processor

BACKGROUND OF THE INVENTION

The invention relates to a processor according to the preamble of Claim 1 hereafter appended. At present, there is a trend in circuitry design towards building a so-called **D**igital **V**ideo **P**latform (**DVP**) that will perform various multimedia processing functions. Such functions may be effected in hardware, in software, or in a mixture thereof, such choice depending on the processing function itself, and/or on the manufacturing volume of the function and/or circuit in question. The multimedia may include video, graphics, audio, or other.

For reasons of economy, quite often such processor will be dedicated to the execution of only a limited subset of those functions, often even to executing only a single one function. This policy will render a shared bus that connects the various processors to a background memory a key facility of an overall processing system. Now, for controlling the overall system, often furthermore a **C**entral **P**rocessing **U**nit (**CPU**) will be provided. Next to controlling the background memory, the CPU may immediately access various control registers in the various processors. The number of such processors in realistic systems may have risen to 10-20.

The present invention is directed to solving a problem that has been recognized when designing a multi-processor coprocessor that is able to perform both **M**otion **E**stimation (**ME**) and **M**otion **C**ompensation (**MC**). In a complex system like this, the prevailing bandwidth on the shared bus is a prime design issue, and the various processors should maintain synchronization on the time slot level of the processing of an entire field or frame.

SUMMARY TO THE INVENTION

In consequence, amongst other things, it is an object of the present invention to allow programmable slowdown of one or more of the processors being effected in a straightforward manner. Now therefore, according to one of its aspects the invention is characterized according to the characterizing part of Claim 1. The inclusion of stalling cycles

will appreciably lower bus load, leaving free the remainder of the bus capacity that may be applied to other purposes.

Advantageously, the programming means are arranged according to Claim 7. This is a straightforward and hardware-efficient solution.

BRIEF DESCRIPTION OF THE DRAWING

These and further aspects and advantages of the invention will be discussed more in detail hereinafter with reference to the disclosure of preferred embodiments, and in particular with reference to the appended Figures that show:

Figure 1, a general block diagram of a video processing system;

Figure 2, a multiprocessor chip embodying the present invention;

Figure 3, a programmable video processor according to the present invention;

Figure 4, an embodiment of a programming accumulator.

Figure 5, a Table showing Highway Transfer Data for a standard-size scalable pixel block;

Figure 6, a further Table showing Data Rates during ME/MC for implementing such scalability.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Figure 1 illustrates a general block diagram of a video processing system. In this conceptual arrangement, signal sources, and in particular, video sources 42, 44, will present video images for processing onto input communication facility 41 that may be a bus or another sharing organization among various stations. Item 20 is a processing chip, which will be discussed more in detail hereinafter, and which will process the images as received. To this effect, chip 20 is associated to RAM 22 that may store an appropriate amount of information to smoothingly cope with peak flows from sources 42, 44, and as the case may be, with peak requests from video users 46, 48. The latter will use video images as having been processed by chip 20. To this effect, items 20, 46, 48, are mutually interconnected through output communication facility 45 that may be a bus or may be sharing among stations in another manner.

Figure 2 illustrates a multiprocessor chip that is arranged for executing the processing and therewith embodying the present invention. Apart from the Random Access Memory 22, the remainder of the Figure has been compacted into a Single Solid State chip 20. Within this chip, interfacing between bus facility or Onchip Data HighWay 28 and

memory 22 is by way of Main Memory Interface 24 and Bus Arbiter 26. Further bus-connected subsystems are Video Input Interface 30, Memory Based Scaler 32, Video Output Interface 34, Central Processing Unit 38 and Processor 36 that executes both Motion Estimation and Motion Compensation. By themselves, M.E. and M.C. are common features of processing a multi-image sequence such as a film or animation, and the associated procedures will not be discussed herein for reasons of brevity. The same applies to the overall image processing functionality to be provided by processor 36 and the hardware and software facilities necessary therefor.

For discussing the relevance of the data transfer on the bus facility, various modes of use will be considered. Now, the processor 36 may operate in a time-multiplexed manner on ***three*** prime tasks. First, it calculates the motion vectors of an applicable image (ME), then it performs motion compensation on the luminance signal (MC-Y), and finally, it performs motion compensation on the chrominance signal (MC-UV). In principle, the processing block in question may handle an image of arbitrary size, but in the embodiment the maximum throughput is two video streams of 512*240 pixels at 60 Hz, or alternatively 512*288 pixels at 50 Hz. A particular standardized stream amounts to 720*240 pixels at 60 Hz, or alternatively, 720*288 pixels at 50 Hz.

Examples of use are defined by various operational parameters. The actual ***display mode*** determines which conversion must be executed, which is usually a fixed property of a particular video product once it has been designed, inasmuch as changing of the display scan format is often unviable. The display mode has the following parameter values for determining the actual conversion. Note that the selecting and management among all of these cases is controlled by the CPU, and some of these selection and management functionalities may even be changed dynamically, during run-time.

Applicable data rates are as follows

- 50 i / 60 i = 1 times the data rate
- 100 i / 120 i = 2 times the input data rate;
- 100 p / 120 p = 4 times the input data rate.

The ***scalability mode*** allows the application to effect a trade-off between image quality and the amount of resources used, such as highway bandwidth and available amount of background memory. This effectively controls the quality attained versus the resources that are availble.Various possibilities are as follows:

- frm - fld – fld , previous frame, current field, and next field;
- frm - fld , previous frame and current field .

- fld - fld , previous field and current field

The ***data mode*** controls the amount of video that must be processed, such as only one main widow, as distinct from a background combined with a picture-in-picture display. Various possibilities are:

- one “standard” stream of 720 pixels width
- two “small” streams of 512 pixels width
- Anything else that may lie within the maximum supported image size

The block 36 has been designed in the embodiment with the following properties:

- Motion estimation requires 1024 cycles to process 128 x 8 pixels
- Motion compensation requires 1600 cycles to process 128 x 8 pixels
- The clock frequency is 150 MHz.

Figure 3 illustrates a programmable video processor according to the present invention. Within processor 50 there is an interface for communicating with other subsystems such as those shown in Figure 2. Internal communication is effected by internal local bus 60. The various stations or facilities connected thereto are program ROM 54, programmable PROM 54 for storing program and/or data, data RAM 58, and finally processing element 56 that has both input and output coupled to the local bus 60. Various control, address, and data interconnection lines have been ignored for brevity, inasmuch as they would represent straightforward solutions to persons skilled in the art.

Figure 4 illustrates an embodiment of a programming accumulator. Herein, a programming register 72 is loaded via line 70 with a first number. Under clock synchronization, the register content is forwarded to adder 74 for addition to the content of accumulator register 76, the content being retrocoupled through interconnection 78. The sum of the two data is written back to accumulator storage facility 76. Now, the higher the content of register 72, the more frequently carry output 80 from accumulator 76 will generate a carry signal. The carry signal will then control an effective clock cycle for therewith having execute the processor of the present invention an image processing operation.

Figure 5 is a Table showing Highway Transfer Data for a standard-size scalable pixel block of 128 X 8 pixels, during motion estimation and motion compensation for the various display modes. Motion estimation and motion compensation require approximately the same input data but produce different output data, and also different amounts of output data. Clearly, the total variation is about + 50% in the rightmost column.

Figure 6 is a further Table showing Data Rates during ME/MC for such scalability, and in particular, the consequences arising for the highway bandwidth during ME and MC for the various display modes recited supra. In a typical system, the memory is operating at 166 MHz, 32 bits dual data rate, which results in a theoretical maximum highway bandwidth of

(166 * 2 * 4) or approximately 1200 Mbyte/sec.

During ME, the throughput requirement is 732 Mbyte/sec. This bandwidth should therefore in principle being continually available, even in a relatively slow 50i/60i system. On the other hand, one would wish that such relatively slow system should be able to operate at a lowered data rate in comparison with the modes requiring higher display rates. In fact, one should wish to relinquish a certain amount of bandwidth, at a cost of a few extra clock cycles. In consequence, the present invention offers a ***programmable*** slow down facility, inasmuch as the optimum would depend on the actual display mode. A further requirement is to have the present invention introduce a facility to save bandwidth also for the processing of smaller images.

The present invention will therefore offer a programmable slowdown factor in the digital circuitry of the coprocessor. For a slowdown factor of S, that is any real number, ≥ 1, the following holds:

- Motion Estimation requires S * 1024 cycles to process 128 * 8 pixels;
- Motion Compensation requires S * 1600 cycles to process 128 * 8 pixels.

On the basis of the software governing the display motion, the slowdown factor will be easily set in this manner. An advantageous embodiment is through an accumulator that periodically accumulates an appropriate operand. The carry output will rise to high whenever the accumulator overflows. The carry out will be controlled by the overflows/wraps, for thereby controlling the stalling of the overall processor. Giving a few embodiments hereinafter for Motion Estimation would render the presenting of similar measures for Motion Compensation superfluous.

For a value of S = 1.215, we want 1024 * 1.215 = 1244 cycles to compute 128 * 8 pixels. That means that we want stalling 1244 – 1024 = 220 times in a 1244 cycle interval. The correct programming would therefor be <u>**x**</u> =220 / 1244 = 0.1768489.

For a value of S = 16, we want 1024 * 16 = 16384 cycles to compute 128 * 8 pixels. That means that we want stalling 16384 – 1024 = 15360 times in a 16384 cycle interval. The correct programming would therefor be <u>**x**</u> =15360 / 16384 = 0.9375. Clearly, **x**

$= (S - 1) / S$. Implementing a long accumulator register will allow accurate programming of the required factor. A 10-bit accumulator has the parameter N to be set by the CPU to control the programmable slowdown: $N = \text{round} (1024 * x)$. For the two factors supra, such will result in the following:

S = 1.215; x = 0.1768489; N = 181.

S = 16; x = 0.9375; N = 960.

A further advantage of the programmable stalling according to the preceding is that it will allow other bus master stations, such as other coprocessors that have a lower priority than memory, to have relatively smaller buffers than would have been the case otherwise. Especially in the interval during which the stalling processor does not access the bus, lower priority master stations will be periodically allowed to temporarily grab the bus. In fact, this feature leads to smaller IC area, and inherently, to lower manufacturing costs.

The above embodiments of the invention have been presented by way of examples, rather than by way of limitation. In consequence, persons skilled in the art will recognize various changes and amendments that would not exceed the scope of the invention, inasfar as such scope has been covered by the appended Claims.